Fraud Detection in E-Commerce

Sara Alqethami[1], Badriah Almutanni[2] and Manal AlGhamdi[3]
maalghamdi@uqu.edu.sa
College of Computer Science and Information System, Umm Al-Qura University, Makkah, Saudi Arabia

Summary
Fraud in e-commerce transaction increased in the last decade especially with the increasing number of online stores and the lockdown that forced more people to pay for services and groceries online using their credit card. Several machine learning methods were proposed to detect fraudulent transaction. Neural networks showed promising results, but it has some few drawbacks that can be overcome using optimization methods. There are two categories of learning optimization methods, first-order methods which utilizes gradient information to construct the next training iteration whereas, and second-order methods which derivatives use Hessian to calculate the iteration based on the optimization trajectory. There also some training refinements procedures that aims to potentially enhance the original accuracy while possibly reduce the model size. This paper investigate the performance of several NN models in detecting fraud in e-commerce transaction. The backpropagation model which is classified as first learning algorithm achieved the best accuracy 96% among all the models.
Key words:
Artificial neural network, fraud detection, e-commerce, Backpropagation, Steepest Descent, Gauss-Newton algorithm, QuickProp.

1. Introduction

The continuous growth in e-commerce and online stores has led to an increase in using Credit Cards (CCs) for online purchases. It increased in the last two years due to the COVID-19 pandemic social distancing and lockdown policies. According to study [1] despite the decline in sales of some commodities and economic losses, there was an increase in online grocery shopping by more than 75% at the end of April 2020, there has been also an increase in e-commerce in general [1]. Using credit CCs frequently for online payment make individuals more vulnerable to online attacks that try to steal their credit information and use it for fraud or leak it. The amount of CC data available on the dark web increased by 153% in 2018 compared to 2017 [2] Fraud losses world-wide amounted to 27.85$ billion in 2018 and are expected to increase 35.67$ billion in 5 years and 40.63$ billion in 10 years [3]. Therefore, it is important to develop efficient algorithms that are capable of detecting fraudulent transactions.

Some studies[4] showed that ANNs perform best among various credit card fraud detection techniques. But the disadvantages of ANNs is that they are so expensive to train and can be easily over trained. In order to reduce their expense, were needed to create a hybrid neural network with several optimization technique. This paper aims to compare between two learning methods to find the best model with high accuracy to detect fraudulent activities in e-commerce transactions, particularly the methods that are based on Artificial Neural Networks (ANNs). In this paper we evaluate and compares the performance of two type of NN learning strategies, the first-order learning method and second-order learning method in detecting e-commerce fraud. As well we provide a brief description of the dataset and the processing stage, describes NN models, research methodology, and finally compares the results with previous works.

This research consists of three stages, in the first stage is predicting fraud in e-commerce transactions using ANNs. In the second stage, four learning methods were used to improve the performance of ANN, two from the first-order group and two from the second order group. In the third and final stage, the learning method with the highest results from each group were enhanced using two different training refinements procedures to improve accuracy and possibly reduce the model size. Fig.1 shows stages of the research.

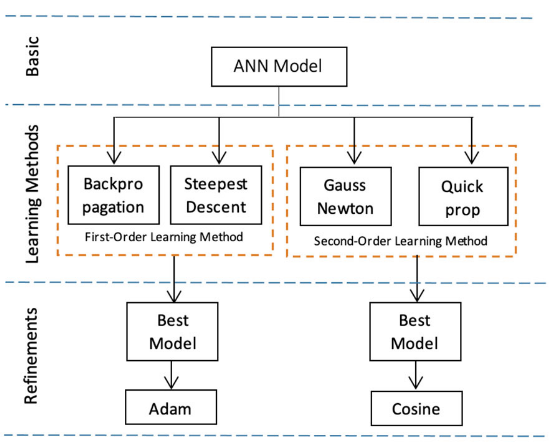

Fig. 1 Stages of the research.

Manuscript received June 5, 2021
Manuscript revised June 20, 2021
https://doi.org/10.22937/IJCSNS.2021.21.6.26

2. Related Works

This section focuses on a brief review for some of the latest related researches. Many papers evaluated the performance of different Machine Learning (ML) methods in detecting CC fraud.

Suresh et al. [5] and Vijay et al. [6] used multiple algorithms of ML such as Naïve Bayes (NB), k-nearest neighbor (KNN), Decision tree (DT), support vector machine (SVM)), logistic regression (LR) and artificial neural network (ANN)were used to predict the occurrence of the CC fraud, where time and amount were used as features and ANNs achieved the best results. Other research proposed by Sun J. et al. [7] investigated the effectiveness of several methods in fraud detection including artificial neural network (ANN), Long Short-term Memory (LSTMs), Recurrent Neural Networks (RNNs), and Gated Recurrent Units (GRUs). Hybrid models were also used to improve the results and the performance of the prediction process by combining more than one ML technique, Douzi et al. [8] implemented Genetic algorithm to increase the enhancement of fraud detection. Panigrahi et al. [9] proposed a hybrid approach of ANNs and Fuzzy Clustering to detect CC fraud. Fuzzy C-Means. A combination of Self-organizing map (SOM) and ANN was proposed by Jain J. et al. [10] to overcome out the limitations of using a single method. Geetha S. et al. [11] and Suharjito S. et al. [12] used Synthetic Minority Oversampling Technique (SMOTE) was used with ANNs and several ML methods to balance the distribution of data and increase the performance of data classification.

3. Fraud Detection Based on Optimized ANNs

This paper aims to compare between two learning methods to find the best model with high accuracy to detect fraudulent activities in e-commerce transactions using Artificial Neural Networks (ANNs). We used two type of learning methods the First-order learning method and Second-order learning method. The same dataset was used for all models with the same features, Sklearn function was used to split the test and train using default settings. In the NN model, the number of input layers was 11 input layers before training and after pre-processing step the input layer to 17 input layers. Increasing the number of iterations reduced the training loss, but it was time-consuming, also increasing the size of the hidden layers enhanced the results to some level. Accuracy was used to evaluate the results.

3.1 Neural Network Models

A computational model that works like the neurons in the human-brain. Where each neuron sends specific operations to the next neuron. It takes an input then performs some specific operations then passes the output to the following neuron. We used Neural Network because it has special ability to derive and detect and solve complex problem better than humans or other computer techniques [13]

In general, the loss function in ANN (mean square error function) needs to be minimized by finding optimized values weights of NN [14]. Fig. 2 shows the block diagram of ANN, where optimization techniques include the backpropagation step, and the error is calculated from the targeted and computed output. Optimization in neural network context is the minimization of the objective function, E_p towards the solution for minimal error value [14]

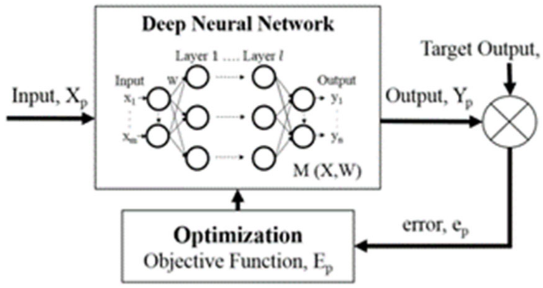

Fig. 2 Block Diagram of ANN [13].

There are two basic categories of optimization methods:

3.2 First-order learning method

First-order learning method provide the capability to deal with structured and multi-relational knowledge. many applications include first order knowledge discovery, induction of integrity constraints in different databases, different predicate learning, and learning mixed theories of predicate definitions [15]. we introduce comparisons between two different models from first-order learning method.

3.2.1 Steepest Descent

Steepest descent method is used for the minimization of a general nonlinear function, also known as the gradient descent method [16].

In this method, the error is decrease along the negative gradient of the error surface, and the learning rate ∈ is applies to all weights and it is adapted internally during training, which start with a bigger value, and halved in each epoch until a value that reduces the error is reached [17] The next table shows the steps of Steepest Descent [16].

Algorithm 1 Steepest Descent Method

Given an initial $x_0, d_0 = -g_0$, and a convergence tolerance tol
for $k = 0$ to $maxiter$ **do**
  Set $\alpha_k = \text{argmin } \phi(\alpha) = f(x_k) - \alpha g_k$
  $x_{k+1} = x_k - \alpha_k g_k$
  Compute $g_{k+1} = \nabla f(x_{k+1})$
  if $||g_{k+1}||_2 \leq tol$ **then**
    converged
  end if
end for

Fig. 3 Steepest Descent Method [17].

the algorithm converges too fast and gives a good accuracy as well as. The fast convergence is due to the continuously changing for the learning rate with respect to error changing.

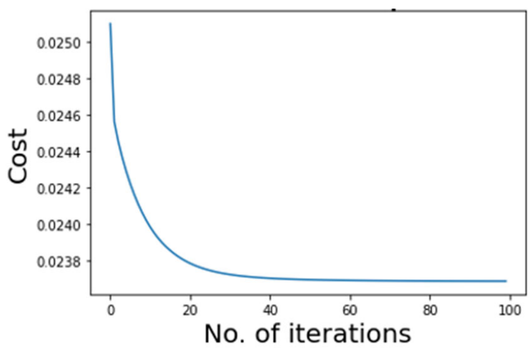

Fig. 4 Cost Function at Steepest Descent.

3.2.2 Backpropagation

The backpropagation algorithm is a gradient descent optimization algorithm (GD) it a type of first-order derivative method that computing the loss function with respect the weight [17]. it begins with random weights and the target is to adjust them to minimize the error to until the neural networks learn the training data set [18].

3.3 Second-order learning method

Second-order derivatives use Hessian to compute the iteration based on the optimization trajectory [14]. Some examples of second-order optimization methods are Newton, conjugate gradient, quasi-Newton, Gauss-Newton, Levenberg-Marqaurdt, and Quickprop.

3.3.1 Gauss-Newton algorithm

The Gauss-Newton algorithm iteratively finds the value of the variables that minimize the sum of squares according to given functions[19]. Gauss-Newton update rules and process the optimization iteratively by updating coefficient values (values needed to solve) using the following Gauss-Newton update rule [19]:

$$\beta^{K+1} = \beta^{K} + J^{+} r(\beta^{k}) \qquad (1)$$

where

β^{K+1} = updated coefficient values.

β^{K} = current estimate for coefficients values.

J^{+} = pseudoinverse of the Jacobian matrix, where $J_{ij} = \partial_{rj}(\beta^{k})/\partial\beta_j$

$r(\beta^{k})$ = residual vector, calculated with the current estimate for coefficients.

The second-order learning method adopts the following general formulation [19].

$$w_{m+1} = w_m + \Delta w_m = w_m - H^{-1} g_m, m \geq 0 \qquad (2)$$

Where (H) Hessian matrix match to the second-order derivative of the error function, which can be approximated by a Jacobian matrix expression, as[19].

$$H = \frac{\partial^2 E}{\partial_{wi}\, \partial_{wj}} \approx J^T J \qquad (3)$$

3.3.2 Quickprop

QuickProp was proposed to speed up the convergence process of backpropagation learning method [17]. and is considered the most basic second-order optimization method [19]. It implicitly uses the curvature and involves the slope of the error surface at a point that determines by the current weights [17], and doesn't use the second-order derivative of the error immediately but instead attempts to get the error curvature approximation out of the sequential rating of first-order error derivatives [19]. The QuickProp update rule is as follows:

$$w_{m+1} = w_m + \Delta w_m = w_m + \frac{g_m \Delta w_{m-1}}{g_m - g_{m-1}} \qquad (4)$$

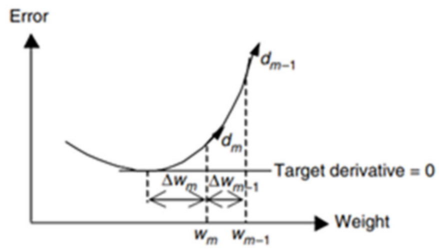

Fig. 5 Illustration of QuickProp learning method that implicitly involves curvature of error surface [17]

3.4. Refinements Methods

Most training refinements procedures aims to potentially enhance the original accuracy while possibly reduce the model size. In this paper two refinements methods were used to improve the models with the highest results.

Two refinements methods were applied to each model separately.

3.4.1 Cosine Learning Rate Annealing

In this method, the training process begin during a specific number of epochs to learn with the full learning rate then began using the cosine function [20]

3.4.2 Adam Optimization Method

Adaptive Moment Estimation (Adam) it is stochastic optimization method that combines between root mean squared prop (RMSprop) and momentum. The algorithms facilitate of adaptive learning rates methods to get individual learning rates for any parameter [20].

4. Experiments

A publicly available dataset was acquired from Kaggle[1] website and was used for training and evaluating the proposed methods. The data contain 151,112 records, where 14,151 records are classified as fraud and the rate of fraud data is 0.093%. The likelihood of fraud per category is showing in Fig. 3 shows us each source (Ads, Direct Source, SEO), browser (Chrome, FireFox, IE, Opera, Safari), and sex (male, female). In Fig.4 shows the overview of distribution by country of origin.

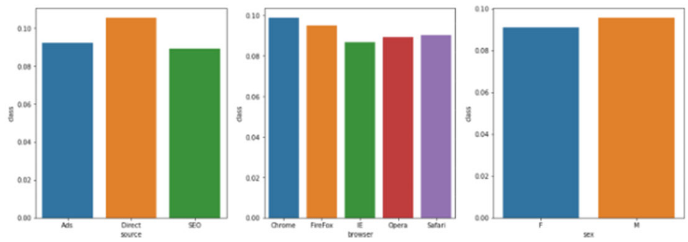

Fig. 6 The likelihood of fraud per category.

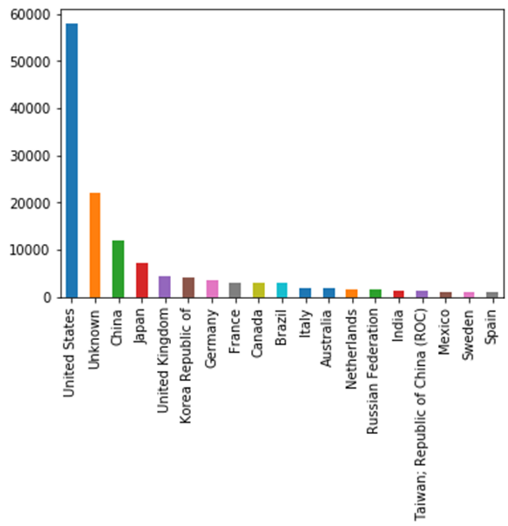

Fig. 7 The overview of distribution by country of origin.

4.1. Pre-processing

The dataset consisted of two files holding the following data:

4.1.1. Fraud_data.csv file:

Contains basic information about the user and the payment including user ID, gender, age, signup time, purchase time and amount, IP address, browser type, and the class of each purchase (fraud or non-fraud).

4.1.2. IpAddress_to_Country.csv file:

Contains the countries generated from a combination of the lower bound and upper bound of an IP address.

First step to prepare the data was merging the two files into one file called "Fraud data with country.csv". The final dataset has the following categorical variables: source, browser, sex, IP country, purchase month, purchase dow and the following continuous variables: purchase value, age, device id freq, quick purchase, and countries from device.

[1] https://www.kaggle.com/vbinh002/fraud-ecommerce

For pre-processing, all the continuous variables were normalized. Dealing with continuous variables is very straight forward in neural network, categorical variables on the other hand need special handling. Here we use the concept of categorical embedding and create an embedding layer for any categorical variable. The number of embedding vectors of a variable is equal to the number of its unique values plus one (for unknown values of the variable). The embedding vector dimension is computed using the following formula:

$$dim = 1.6 * n^{0.56} \quad (5)$$

Where n is the number of unique values of the categorical variable. Finally, we split the data in training (70%), validation (15%) and testing (15%) splits.

4.2. Feature Engineering

We add columns that can be good indication of fraud. We added the following columns:

- Device id freq: how many times a device id appeared in the data.
- Countries from device: the number of different countries.
- Quick purchase: 1 if the difference between purchase time and signup time is less than 30 second and 0 otherwise.

We also extract temporal features from the dataset:

- Purchase month: the month of the purchase extracted from purchase time.
- Purchase dow: the day of week of the purchase extracted from purchase time.

Finally, we drop useless columns: user id, ip address, signup time, purchase time, device id.

The dataset is heavily imbalanced with only 9.36% belonging to the positive class (fraud).

4.3. Performance Metrics

For evaluation, the following measurements were used:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (6)$$

$$Sensitivity\ (Recall) = \frac{TP}{TP + FN} \quad (7)$$

$$Sensitivity\ (Recall) = \frac{TP}{TP + FN} \quad (8)$$

$$F_1 = \frac{TP}{TP + 1/2(FP + FN)} \quad (9)$$

Where TP, FP, TN and FN given in Equation 6 to 9 represent the true positives, false positive, true negatives, false positive, false negatives, respectively.

4.4. Experimental Setup

For implementation, Python and Jupiter notebook were used with Anaconda navigator. The following python libraries were used to load data, build, train and evaluate the models: math, collections, pandas, numpy, matplotlib, pyplot, torch, torch.nn, torch.optim, random, fastai.tabular.core, fastai.tabular.data, and fastai.tabular.model.

The dataset was split into training, validation and testing set according to the following percentages, respectively 70%, 15%, and 15%.

4.5. Results

Accuracy results without refinements methods of all models are shown in Table 1, and accuracy results with refinements methods of all models are shown in Table 2, where backpropagation-Adam achieved the highest accuracy with 96% followed by Steepest Descent-Adam with 95%. The backpropagation method especially with Adam's algorithm, it is better than other improvements, it was the fastest in training process, because the Adam optimization algorithm it tends to yield extremely fast results. Where QuickProp-Epoch Model achieved the highest accuracy with 95.5% followed by QucikProp 93%. The Gauss-Newton with Epoch refinement got the worst performance. QuickProp-cosine, Gauss-Newton and Gauss-Newton with cosine refinement achieved the same accuracy 90%. The Quickprop method performed better than Gauss-Newton, it was also the fastest in training process because it doesn't use parameters to adjust, and the error goes down much faster initially [13] . Gauss–Newton needed more time to train because finding the Hessian matrix for large networks trained with a large number of training data can be expensive to calculate and time-consuming [13] In addition, inverting the Hessian matrix can cause numerical instability problems and the methods may not perform satisfactorily [13]. Refinement's procedures Cosine Learning Rate Decay and Epoch increasing were used to improve the results of NN models Gauss-Newton. Using these refinements with Gauss-Newton didn't improve the results, on the contrary Gauss performing decreased after using epoch refinement. But this method was able to improve the performance of QuickProp.

Table 1: Accuracy results without refinements methods

Model Name	*Without Refinements*
Backpropagation	95.6%
Steepest Descent	95.2%
QuickProp	93.6%

Gauss-Newton	90.5%

Table 2: Accuracy results with refinements methods

Model Name	***With Refinements***
Backpropagation-Adam	96%
Steepest Descent – Adam	95%
QuickProp – Cosine	90.5%
QuickProp – Epoch	95.5%
Gauss-Newton- Cosine	90.5%
Gauss-Newton- Epoch	77.3%

4. Comparison with Related Methods

Many works were conducted to detect to detect fraud transactions using machine learning (ML) methods, but since most of these methods used different datasets and evaluation measurements, it is hard to compare to results objectively. The results of the related works are show in Table 3.

Saputra A. et al[12]. used the same dataset we used in our study, and their results were evaluated using different measurements including accuracy. The highest accuracy was achieved by their Neural Network model without SMOTE 96%, while in our study the backpropagation-Adam model achieved 97%. This is due to the choice of the data processing techniques and the used optimization method.

In our study, we used the first and second-order learning methods to improve the accuracy by improving the way NN train the data, while in Saputra A. et al [12] paper they used SMOTE (Synthetic Minority Oversampling Technique) to improve the accuracy by creating balance data, but the results of SMOTE were not encouraging. It has been observed that in general NN can obtain good results in detecting fraud transactions compared to other methods and these results can be improved using appropriate optimization methods, otherwise the unsuitable optimization method may reduce the performance and the classification accuracy.

Table 3: Comparison with related works

Models	***Dataset***	***Results***	***Reference***
NN+HSA	German dataset	Accuracy 86%	Daliri, S. [21]
Logistic Regression KNNS SVM Decision Tree Random Forest Xgboost	European Dataset	AUROC=0.96% 0.97% 0.97% 0.95% 0.98% 0.98%	Niu X. et al.[22]
Logistic Regression Random Forest SVM	I-Cheng Yeh's Dataset	Accuracy 77% Accuracy 81% Accuracy 65%	kumar, Y [23]
NN NN with SMOTE Random Forest Random Forest-SMOTE Decision Tress Decision Tress-SMOTE Naïve Bayes Naïve Bayes-SMOTE	Public dataset on Kaggle	Accuracy 96% 85% 95% 95% 91% 91.6% 95% 95%	Saputra A. et al. [12]

4. Conclusion

This paper investigated the performance of several optimization methods that belongs to the first-order learning algorithms (Backpropagation, Steepest Descent) and second-order learning algorithms (Gauss- Newton, Quickprop) to improve ANN models in detecting fraud in e-commerce transactions. The dataset was obtained from freely open source, and it was cleaned and processed to get the best features and detecting results. Two different refinements techniques were also applied to enhance the original accuracy. We conclude that it is possible to improve the performance of ANNS in detecting fraud efficiently using these methods but in some cases, it would affect the results negatively. The first-order method backpropagation model achieved the best performance with 96% accuracy.

References

[1] "How the Virus Transformed the Way Americans Spend Their Money - The New York Times." https://www.nytimes.com/interactive/2020/04/11/business/economy/coronavirus-us-economy-spending.html (accessed Jun. 08, 2021).

[2] "Amount of Bank Data for Sale on Dark Web up 135%: Report | Credit Union Times." https://www.cutimes.com/2018/07/13/amount-of-bank-data-for-sale-on-dark-web-up-135-re/ (accessed Jun. 08, 2021).

[3] "Payment Card Fraud Losses Reach $27.85 Billion." https://www.prnewswire.com/news-releases/payment-card-fraud-losses-reach-27-85-billion-300963232.html (accessed Jun. 08, 2021).

[4] H. Harwani, J. Jain, C. Jadhav, and M. Hodavdekar, "Credit Card Fraud Detection Technique using Hybrid Approach: An Amalgamation of Self Organizing Maps and Neural Networks," *International Research Journal of Engineering and Technology*, 2020

[5] A. RB and S. K. KR, "Credit card fraud detection using artificial neural network," *Global Transitions Proceedings*, vol. 2, no. 1, pp. 35–41, Jun. 2021, doi: 10.1016/j.gltp.2021.01.006.

[6] V. N. Dornadula and S. Geetha, "Credit Card Fraud Detection using Machine Learning Algorithms," in *Procedia Computer Science*, 2019, vol. 165, pp. 631–641. doi: 10.1016/j.procs.2020.01.057.

[7] A. Roy, J. Sun, R. Mahoney, L. Alonzi, S. Adams, and P. Beling, "Deep learning detecting fraud in credit card transactions," in *2018 Systems and Information Engineering Design Symposium, SIEDS 2018*, Jun. 2018, pp. 129–134. doi: 10.1109/SIEDS.2018.8374722.

[8] I. Benchaji, S. Douzi, and B. el Ouahidi, "Using genetic algorithm to improve classification of imbalanced datasets for credit card fraud detection," in *Lecture Notes in Networks and Systems*, vol. 66, Springer, 2019, pp. 220–229. doi: 10.1007/978-3-030-11914-0_24.

[9] T. K. Behera and S. Panigrahi, "Credit Card Fraud Detection: A Hybrid Approach Using Fuzzy Clustering & Neural Network," in *Proceedings - 2015 2nd IEEE International Conference on Advances in Computing and Communication Engineering, ICACCE 2015*, Oct. 2015, pp. 494–499. doi: 10.1109/ICACCE.2015.33.

[10] H. Harwani, J. Jain, C. Jadhav, and M. Hodavdekar, "Credit Card Fraud Detection Technique using Hybrid Approach: An Amalgamation of Self Organizing Maps and Neural Networks," no. July, pp. 5071–5075, 2020.

[11] V. N. Dornadula and S. Geetha, "Credit Card Fraud Detection using Machine Learning Algorithms," in *Procedia Computer Science*, 2019, vol. 165, pp. 631–641. doi: 10.1016/j.procs.2020.01.057.

[12] A. Saputra and Suharjito, "Fraud detection using machine learning in e-commerce," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 9, pp. 332–339, 2019, doi: 10.14569/ijacsa.2019.0100943.

[13] S. Samarasinghe, *Neural Networks for Applied Sciences and Engineering*. Auerbach Publications, 2006. doi: 10.1201/9781420013061.

[14] H. H. Tan and K. H. Lim, "Review of second-order optimization techniques in artificial neural networks backpropagation," in *IOP Conference Series: Materials Science and Engineering*, 2019, vol. 495, no. 1. doi: 10.1088/1757-899X/495/1/012003.

[15] J. R. Quinlan, "Learning first-order definitions of functions," *Journal of Artificial Intelligence Research*, vol. 5, pp. 139–161, 1996, doi: 10.1613/jair.308.

[16] J. C. Meza, "Steepest descent," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 6, pp. 719–722, Nov. 2010, doi: 10.1002/wics.117.

[17] S. Samarasinghe, *Neural Networks for Applied Sciences and Engineering*. Auerbach Publications, 2006. doi: 10.1201/9781420013061.

[18] M. Puig-Arnavat and J. C. Bruno, "Artificial Neural Networks for Thermochemical Conversion of Biomass," in *Recent Advances in Thermochemical Conversion of Biomass*, Elsevier Inc., 2015, pp. 133–156. doi: 10.1016/B978-0-444-63289-0.00005-3.

[19] T. Huang, Z. Zeng, C. Li, and C. S. Leung, Eds., *Neural Information Processing*, vol. 7667. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. doi: 10.1007/978-3-642-34500-5.

[20] J. Duchi JDUCHI and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization * Elad Hazan," *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[21] S. Daliri, "Using Harmony Search Algorithm in Neural Networks to Improve Fraud Detection in Banking System," *Computational Intelligence and Neuroscience*, vol. 2020, 2020, doi: 10.1155/2020/6503459.

[22] X. Niu, L. Wang, and X. Yang, "A Comparison Study of Credit Card Fraud Detection: Supervised versus Unsupervised." Accessed: Jun. 08, 2021. [Online]. Available: www.aaai.org

[23] Y. Kumar, S. Saini, and R. Payal, "Comparative Analysis for Fraud Detection Using Logistic Regression, Random Forest and Support Vector Machine," *SSRN Electronic Journal*, Mar. 2021, doi: 10.2139/ssrn.3751339.

Sara Alqethami received her bachelor's degree in Computer Sci ence from Umm Al-Qura University, Saudi Arabia in 2018. Curre ntly, she is a master student in Artificial Intelligence in Umm Al-Qura. Her research interests include machine learning and deep le arning and computer vision, and security.

Badriah Al Mutanni is currently master student of Artificial Inte lligence at Umm Al-Qura University. Her academic projects in B aclorios degree include Teaching children with learning difficulti es through educational games uploaded on a website. She is curre ntly working at Investment Administration of UQU.

Manal Algamdi received her Ph.D. degree in Computer Vision from University of Sheffield, United Kingdom in 2015. Her study involved video representation and video similarity measurements. Currently, she is an Associate Professor at the Department of Computer Science, UQU, Saudi Arabia. Manal's research interests include deep learning and its application in different areas of computer vision including HealthCare applications.